Efficient Techniques for Tree Similarity Queries[1]

Nikolaus Augsten

Database Research Group
Department of Computer Sciences
University of Salzburg, Austria

July 6, 2017

Austrian Computer Science Day 2017 / IMAGINE 17

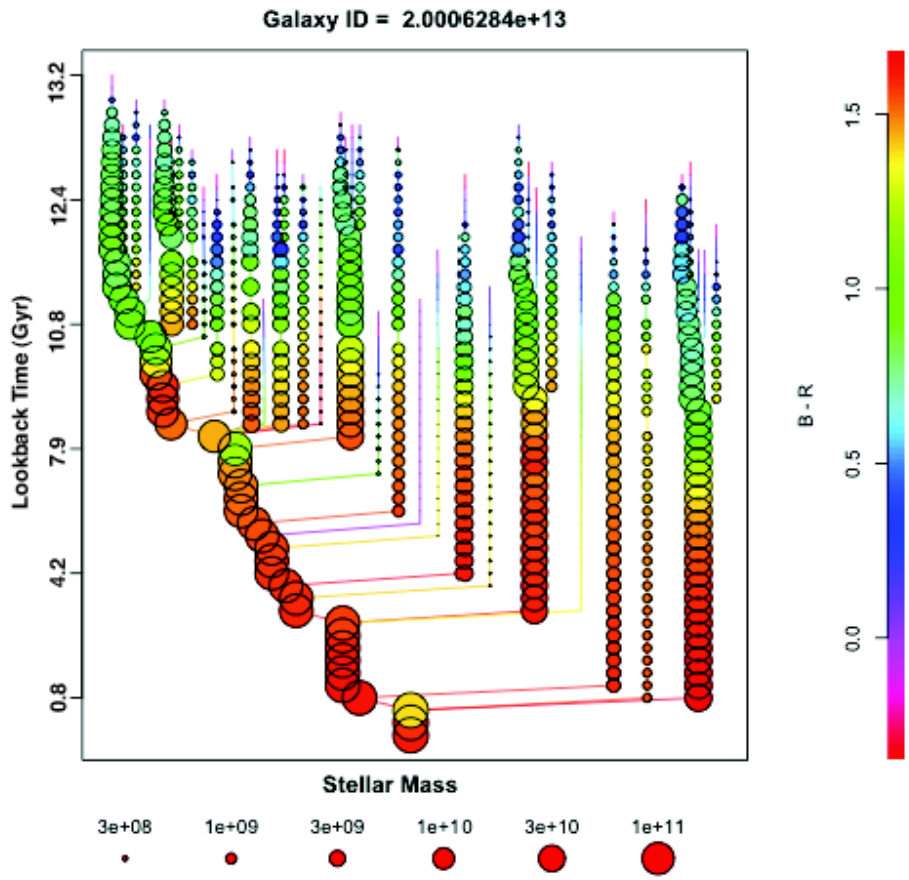
database research group

UNIVERSITY of SALZBURG

[1] Partially funded by Austrian Science Fund (FWF), FFTED Project P 29859

Trees Are Everywhere

- data formats (XML, JSON), directory hierarchies
- ERP: human resources, enterprise assets, bills of material
- NLP: syntax trees of natural languages
- software evolution: source code
- bioinformatics: RNA secondary structure, carbohydrates, neuronal morphology, phylogenetic trees
- image recognition: gesture and shape recognition
- astrophysics: ancestry trees of galaxies (merger trees)

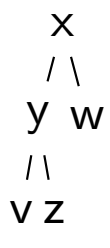
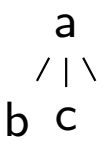
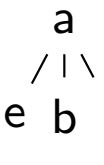


Source: `www.inspirehep.net`

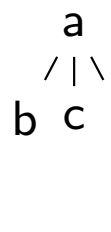
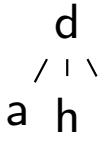
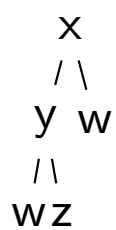
(19M trees, total of 750M nodes)

Similarity Join on Trees

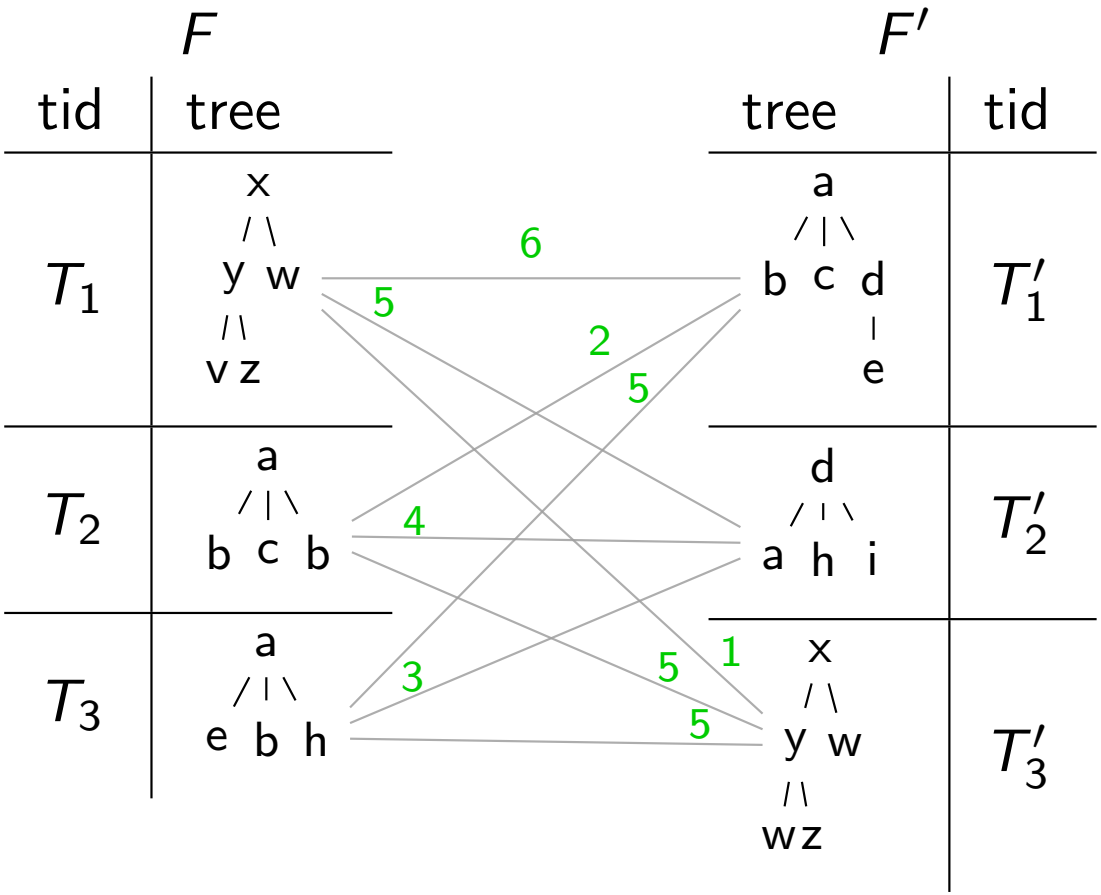
F

tid	tree
T_1	
T_2	
T_3	

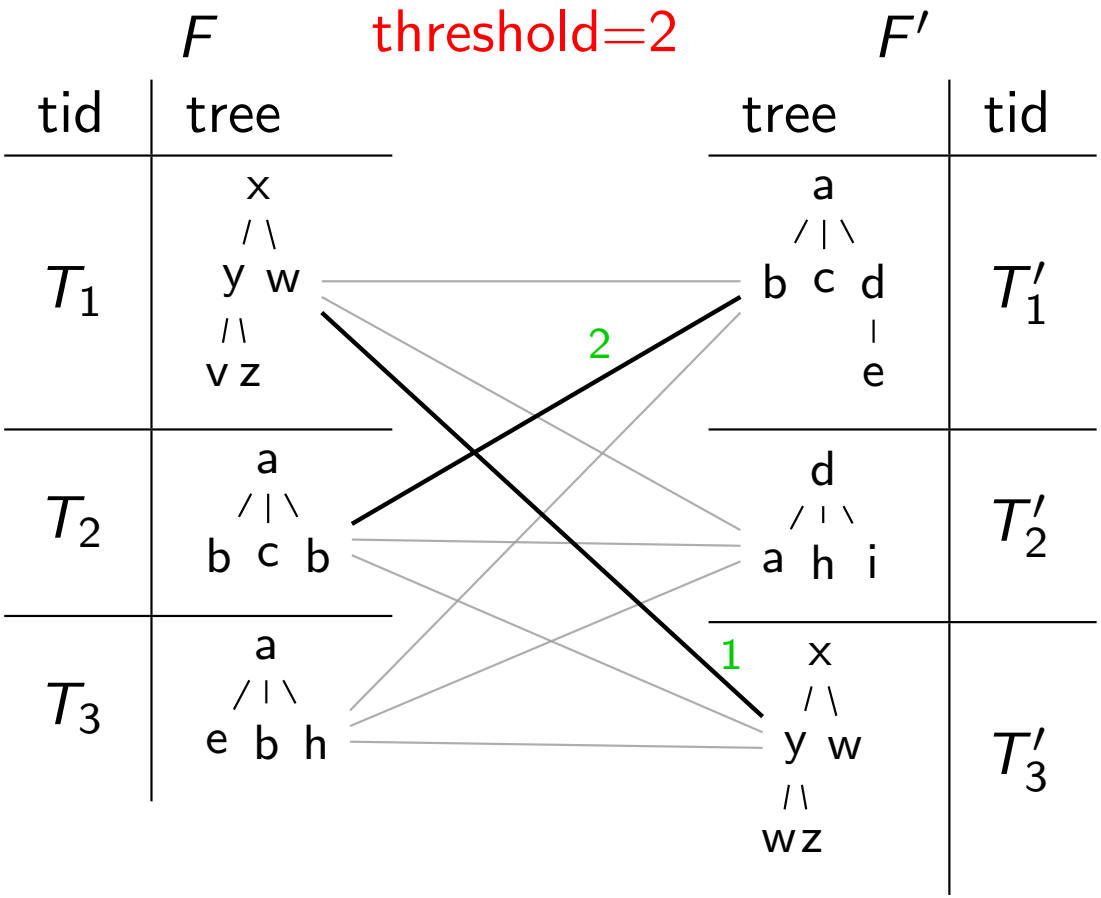
F'

tree	tid
	T'_1
	T'_2
	T'_3

Similarity Join on Trees



Similarity Join on Trees



Outline

1. How similar are two trees?
2. From trees to integer sets
3. Avoiding the nested loop join

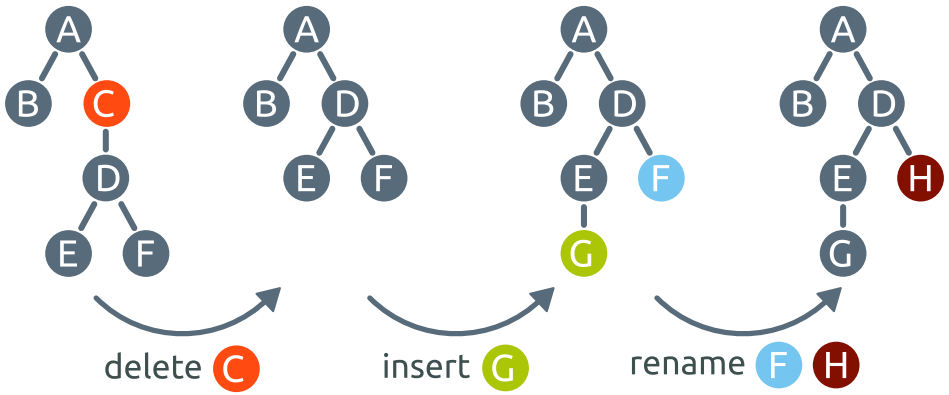
Outline

Tree Edit Distance (TED)

- minimum number of edit operations to transform one tree into another

Tree Edit Distance (TED)

- minimum number of edit operations to transform one tree into another



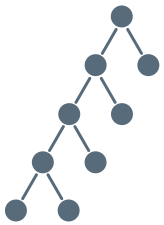
$$\text{TED} = 3$$

State of the Art in TED

Algorithm		Time	Space	Comments
Zhang&Shasha	1989	$O(n^4)$	$O(n^2)$	$O(n^2 \log^2 n)$ runtime for balanced trees
Demaine et al.	2009	$O(n^3)$	$O(n^2)$	worst case is frequent

State of the Art in TED

Algorithm		Time	Space	Comments
Zhang&Shasha	1989	$O(n^4)$	$O(n^2)$	$O(n^2 \log^2 n)$ runtime for balanced trees
Demaine et al.	2009	$O(n^3)$	$O(n^2)$	worst case is frequent

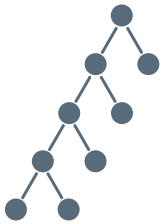


Zhang	0.5 sec
Demaine	30.9 sec

($\approx$ 1000 nodes)

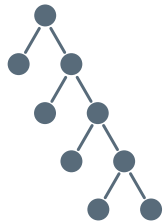
State of the Art in TED

Algorithm		Time	Space	Comments
Zhang&Shasha	1989	$O(n^4)$	$O(n^2)$	$O(n^2 \log^2 n)$ runtime for balanced trees
Demaine et al.	2009	$O(n^3)$	$O(n^2)$	worst case is frequent



Zhang	0.5 sec
Demaine	30.9 sec

($\approx$ 1000 nodes)

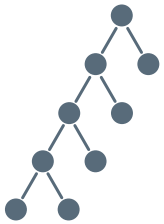


Zhang	247.6 sec
Demaine	26.0 sec

($\approx$ 1000 nodes)

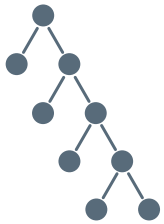
State of the Art in TED

Algorithm		Time	Space	Comments
Zhang&Shasha	1989	$O(n^4)$	$O(n^2)$	$O(n^2 \log^2 n)$ runtime for balanced trees
Demaine et al.	2009	$O(n^3)$	$O(n^2)$	worst case is frequent



Zhang	0.5 sec
Demaine	30.9 sec

($\approx$ 1000 nodes)

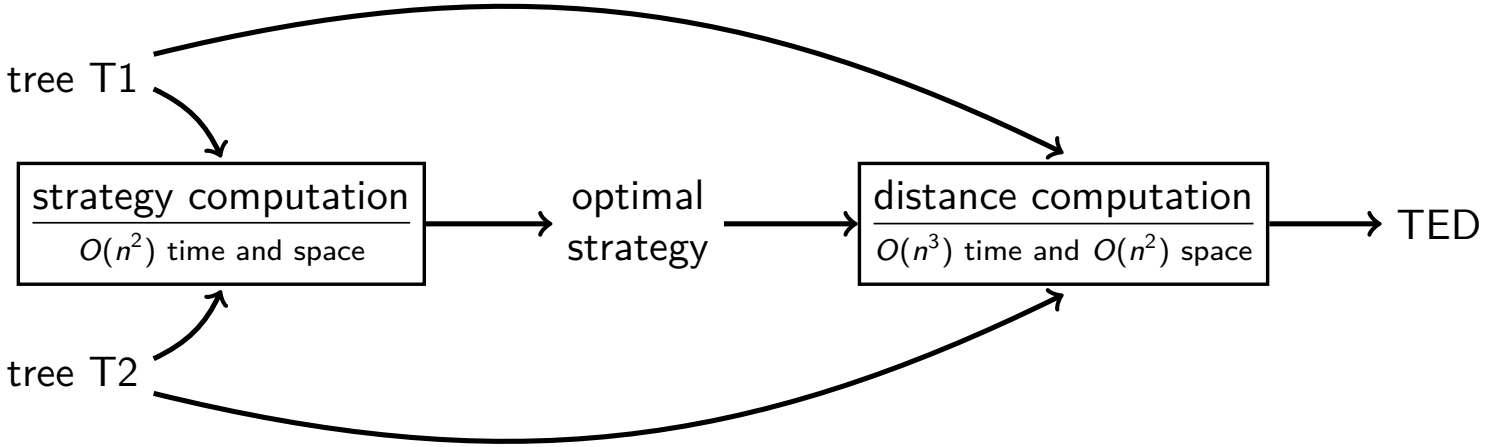


Zhang	247.6 sec
Demaine	26.0 sec

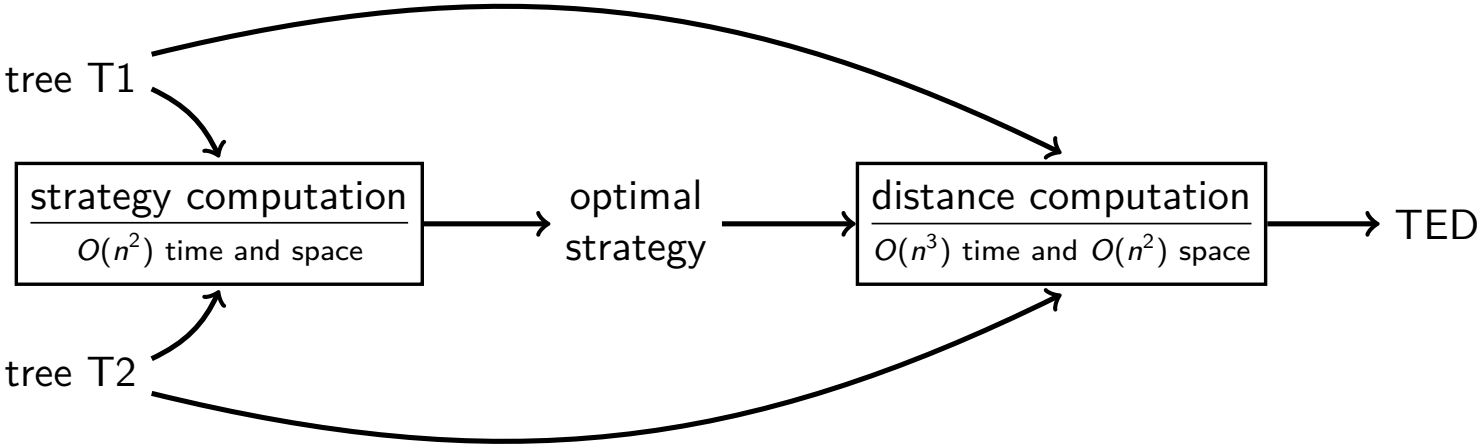
($\approx$ 1000 nodes)

Problem: Hard-coded decomposition strategies

Robust Tree Edit Distance Algorithm (RTED)



Robust Tree Edit Distance Algorithm (RTED)



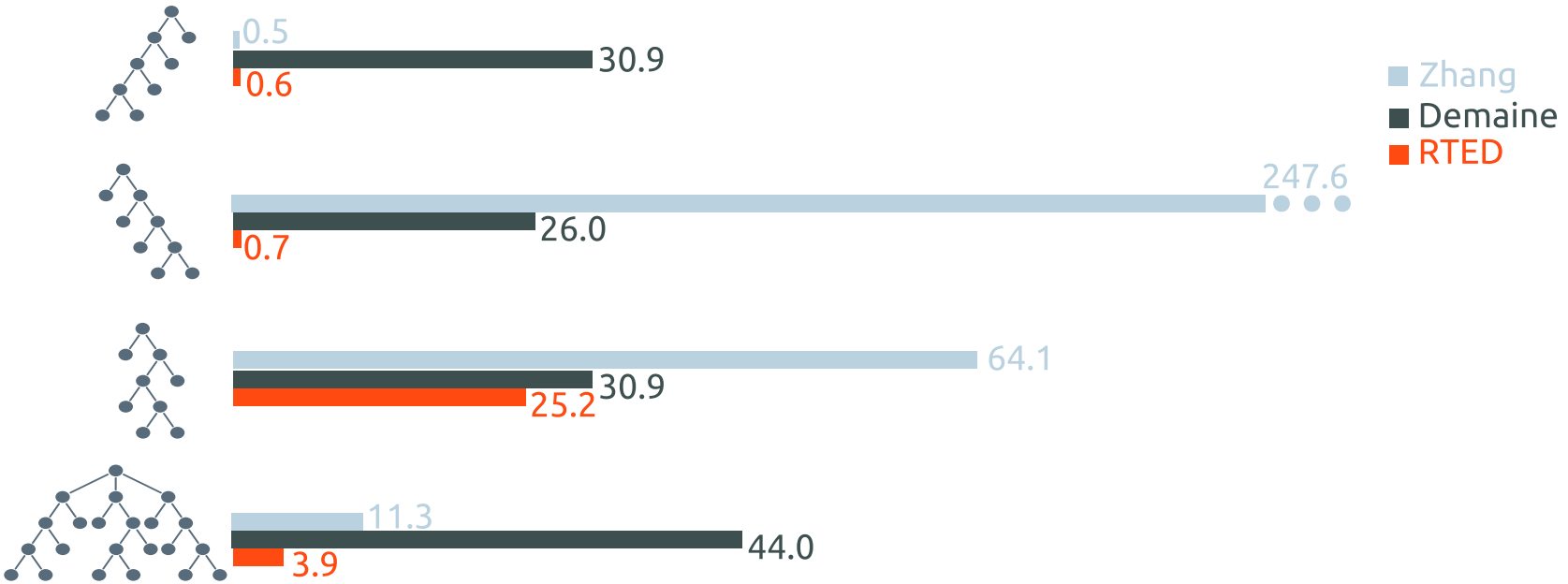
Algorithm		Time	Space	Comments
Zhang&Shasha	1989	$O(n^4)$	$O(n^2)$	$O(n^2 \log^2 n)$ runtime for balanced trees
Demaine et al.	2009	$O(n^3)$	$O(n^2)$	worst case is frequent
Pawlik&Augsten (RTED)	2011	$O(n^3)$	$O(n^2)$	optimal strategy

RTED: fastest known algorithm for TED.

Performance Evaluation

Runtime

- measure runtime (seconds)
- vary tree shape
- tree size $\approx$ 1000 nodes

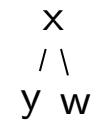
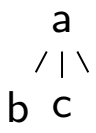
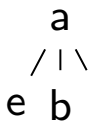


RTED: fastest known algorithm for TED.

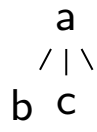
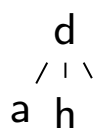
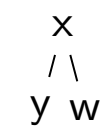
Outline

Similarity Join on Trees

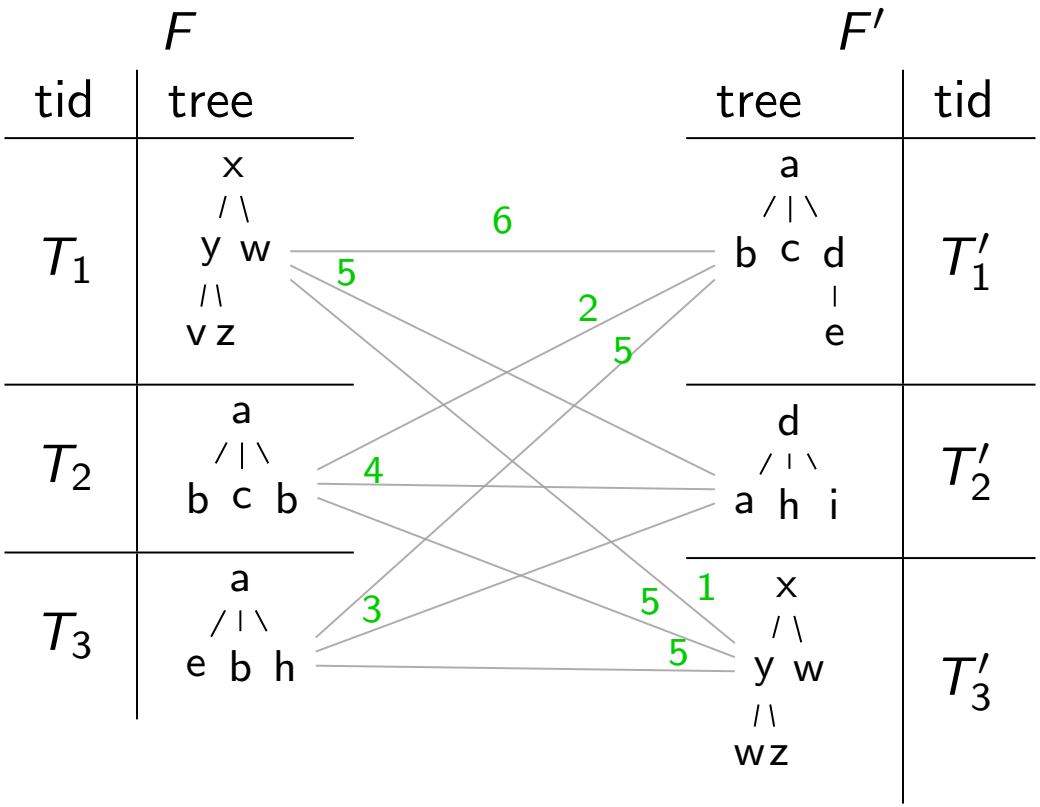
F

tid	tree
T_1	
T_2	
T_3	

F'

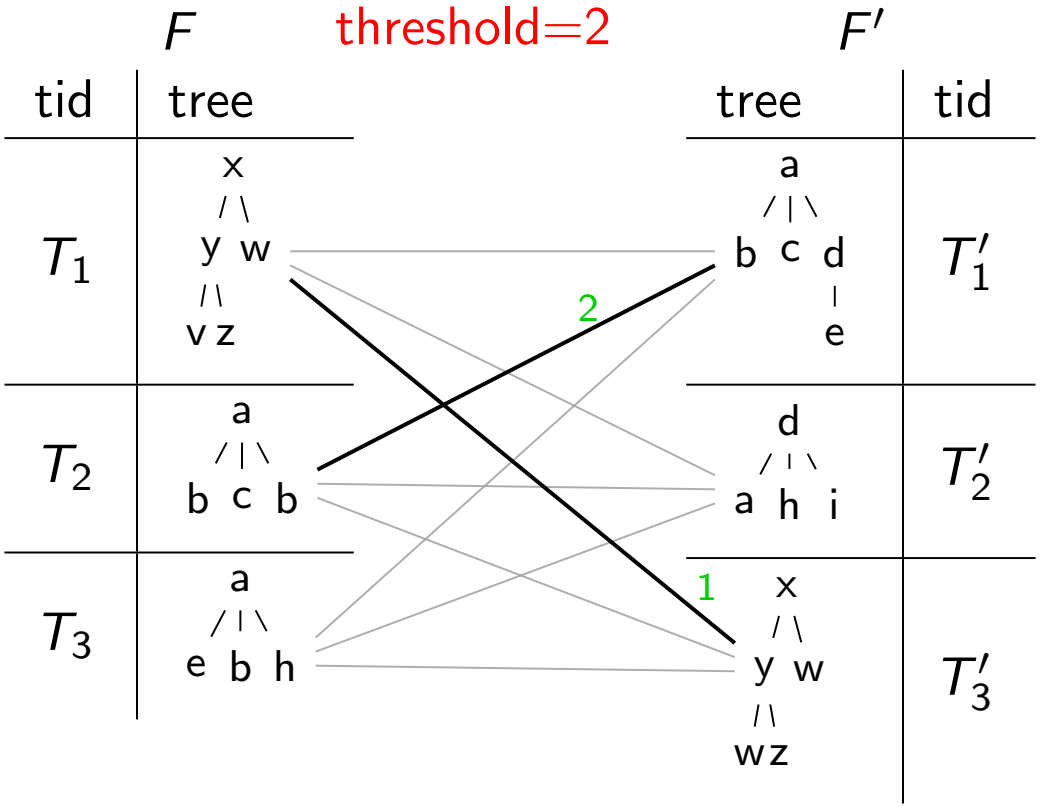
tree	tid
	T'_1
	T'_2
	T'_3

Similarity Join on Trees



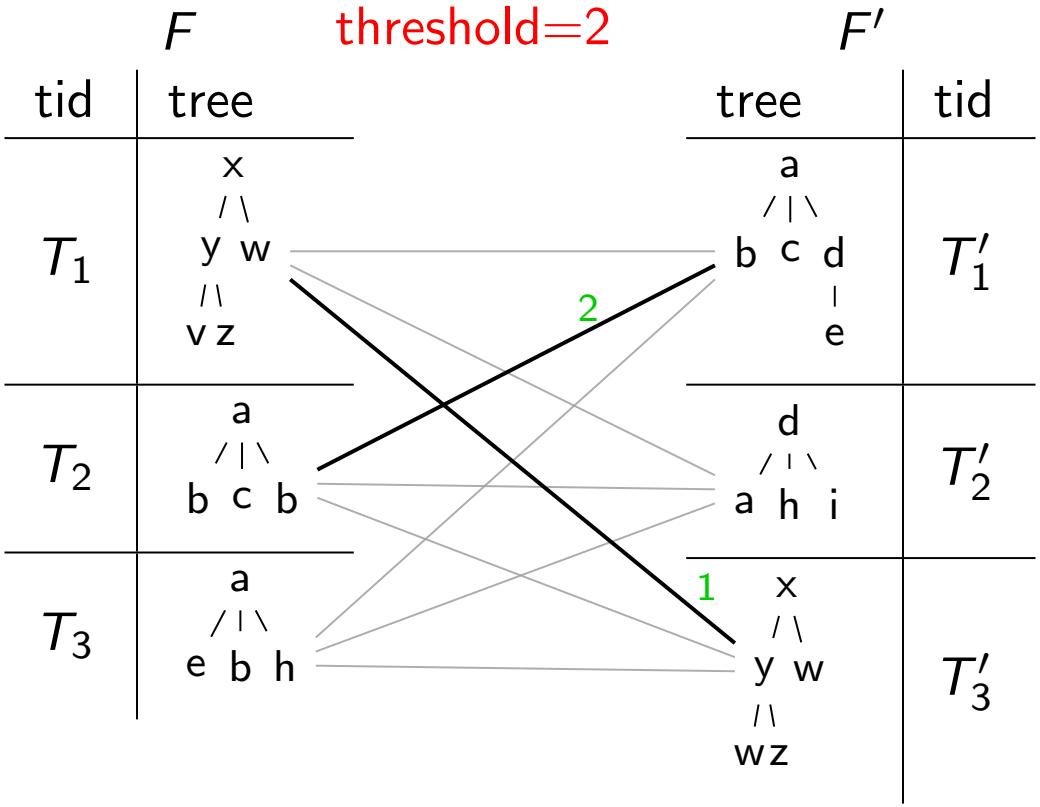
- Nested loop join (NLJ):
  1. compute distance between all pairs of trees

Similarity Join on Trees



- Nested loop join (NLJ):
  1. compute distance between all pairs of trees
  2. return trees within threshold

Similarity Join on Trees



- Nested loop join (NLJ):
  1. compute distance between all pairs of trees
  2. return trees within threshold
- Very expensive: N^2 distance computations!

Standard Join Techniques Fail

- Sorting: Errors destroy sort order

customer

name	city
Frieda	Salzburg
Frodo	Auenland
Maria	Wien

customer

name	city
Frieda	Salzburg
Maria	Wien
Vrodo	Auenland

Standard Join Techniques Fail

- Sorting: Errors destroy sort order

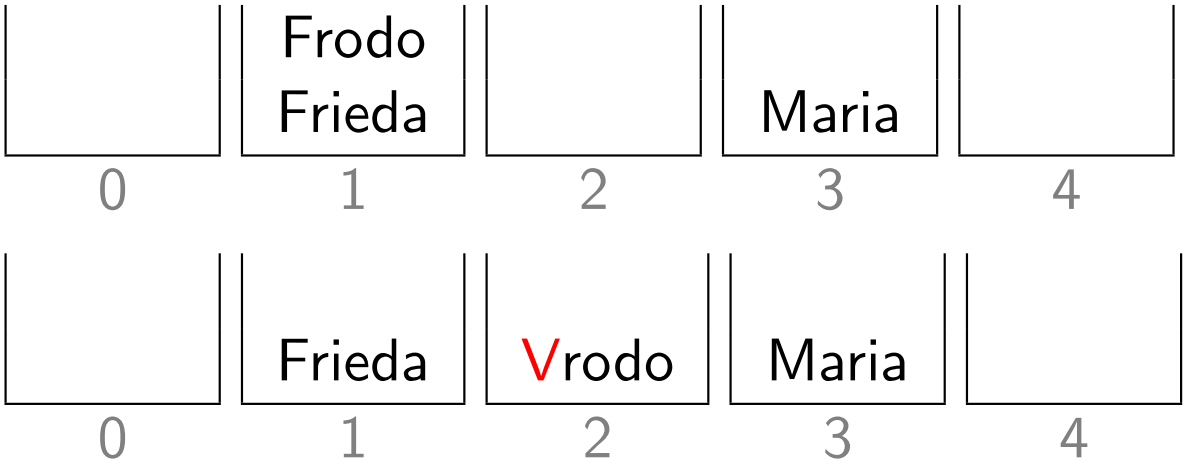
customer

name	city
Frieda	Salzburg
Frodo	Auenland
Maria	Wien

customer

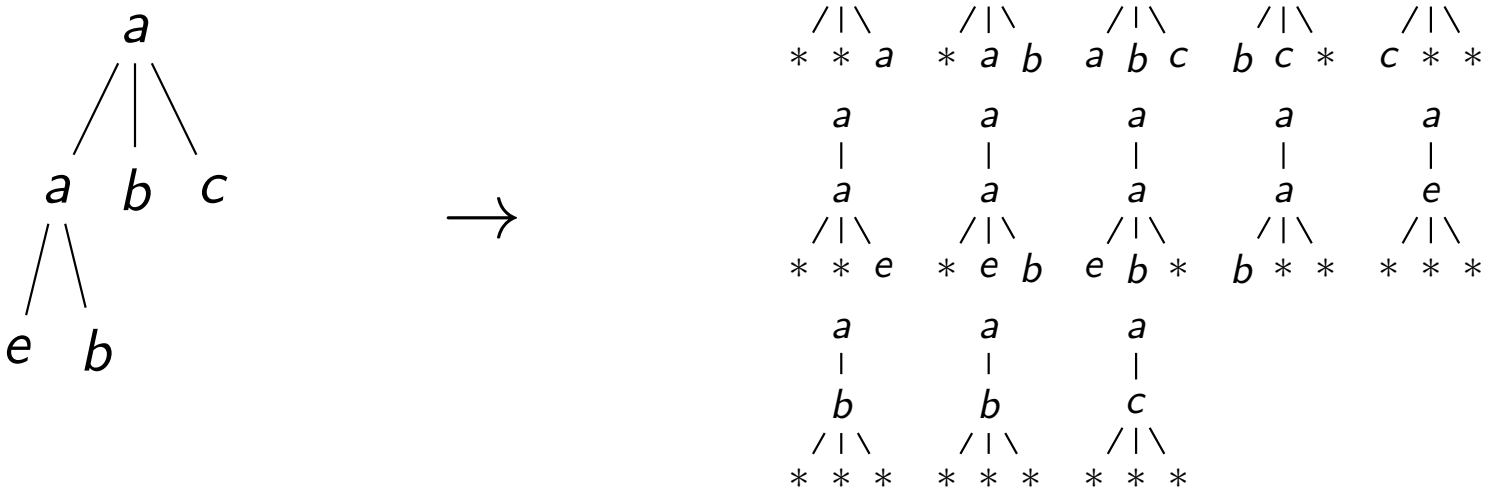
name	city
Frieda	Salzburg
Maria	Wien
Vrodo	Auenland

- Hashing: Errors change hash value



pq-Grams: From Trees to Sets

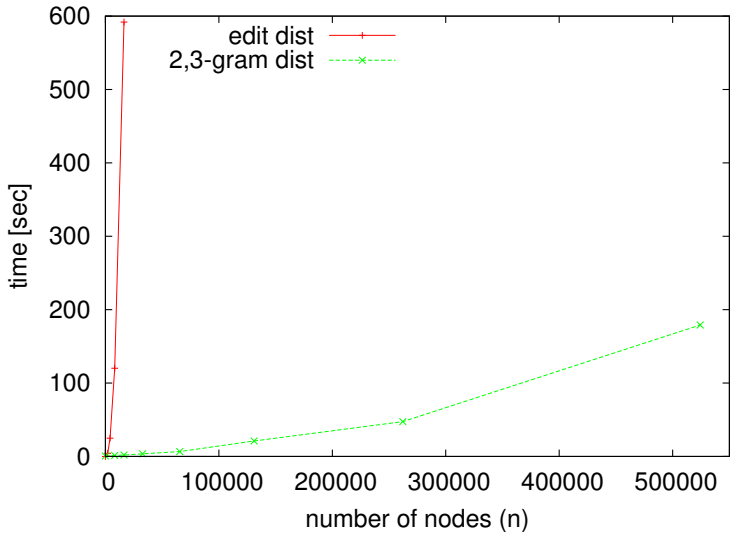
- Split trees into pq-Grams (small subtrees of fixed shape)



- Hash pq-grams to integers
- Intuition: Similar trees many common pq-grams

pq-Grams Have Useful Properties

- Efficiency
  - pq-gram runtime: $O(n \log n)$
  - edit distance runtime: $O(n^3)$
- Filter for edit distance
  - good approximation
  - guarantees **lower bound**
    (fanout weighting of operations)



Tree similarity simplified to set overlap.

Outline

Nested Loop Join for Sets – A Small Experiment

- Dataset: title and author of $n = 873k$ publications (DBLP)
  - avg. 15 words (min: 2, max: 289)
  - 409k different words, Zipfian frequency distribution

Nested Loop Join for Sets – A Small Experiment

- Dataset: title and author of $n = 873k$ publications (DBLP)
  - avg. 15 words (min: 2, max: 289)
  - 409k different words, Zipfian frequency distribution
- Goal: find all similar pairs (e.g. 90% overlap)

Nested Loop Join for Sets – A Small Experiment

- Dataset: title and author of $n = 873k$ publications (DBLP)
  - avg. 15 words (min: 2, max: 289)
  - 409k different words, Zipfian frequency distribution
- Goal: find all similar pairs (e.g. 90% overlap)
- Similarity check is very fast: 6 ns $\approx$ 15 CPU cycles

Nested Loop Join for Sets – A Small Experiment

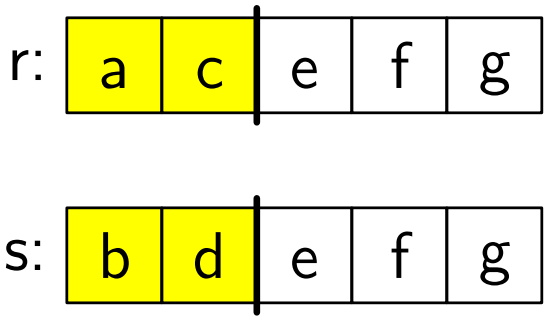
- Dataset: title and author of $n = 873k$ publications (DBLP)
  - avg. 15 words (min: 2, max: 289)
  - 409k different words, Zipfian frequency distribution
- Goal: find all similar pairs (e.g. 90% overlap)
- Similarity check is very fast: 6 ns $\approx$ 15 CPU cycles
- Nested loop join: too many comparisons! (approx. 381 billion)

Overlap	0.95	0.9	0.8	0.7
Runtime [s]	2201	3136	4280	7770

NLJ does not even scale for very fast checks.

Trick: Avoid Touching Hopeless Candidates

- Prefix filter[1]: only look at small part of the set
  - sort sets and cut after prefix
  - no overlap in prefix $\Rightarrow$ similarity threshold not reachable

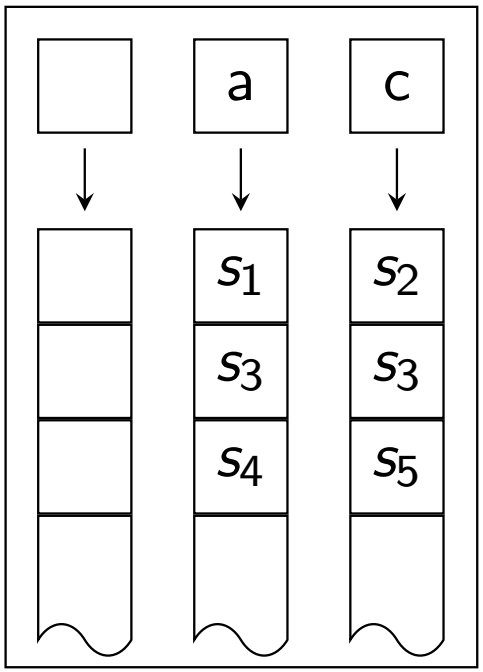


$$|r \cap s| \geq 4$$

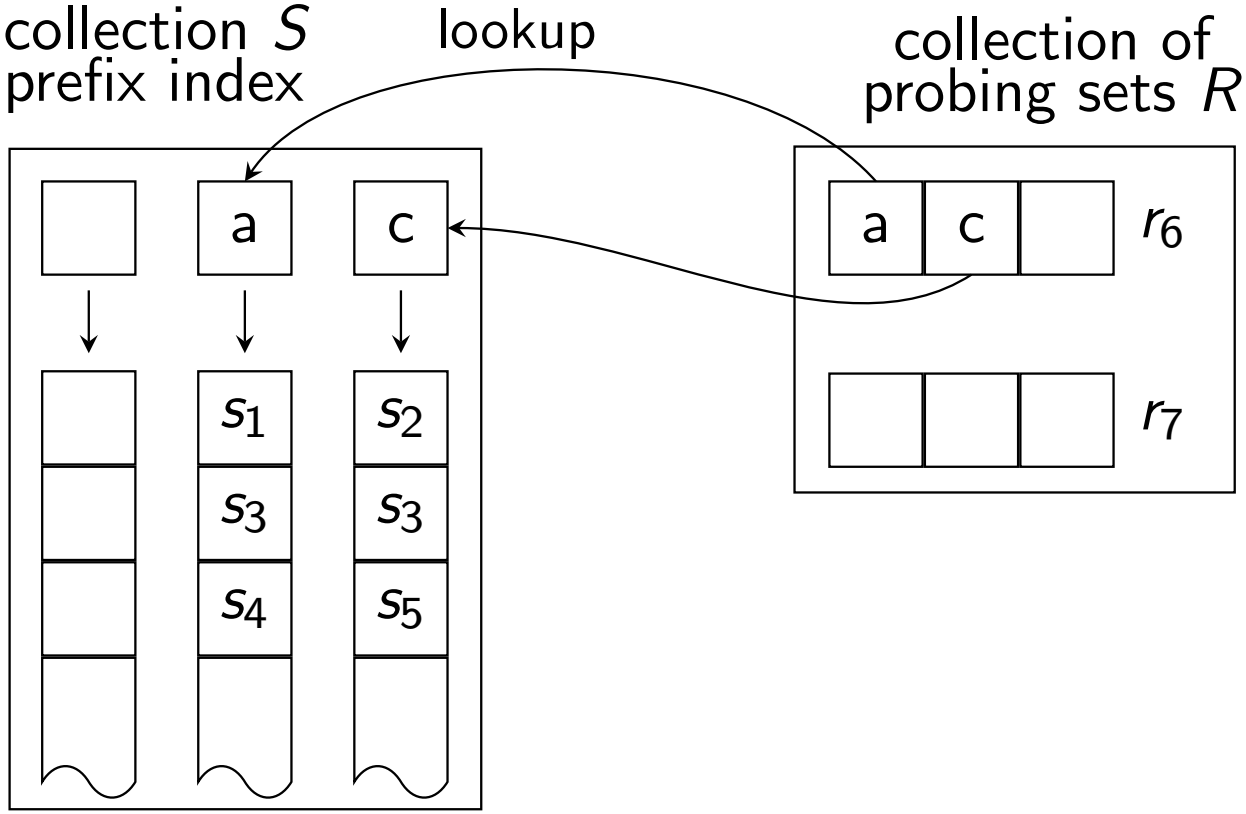
[1] Chaudhuri, Ganti, Kaushik. A primitive operator for similarity joins in data cleaning. ICDE 2006.

Prefix Index Join

collection S
prefix index



Prefix Index Join



Comparison to Nested Loop Join

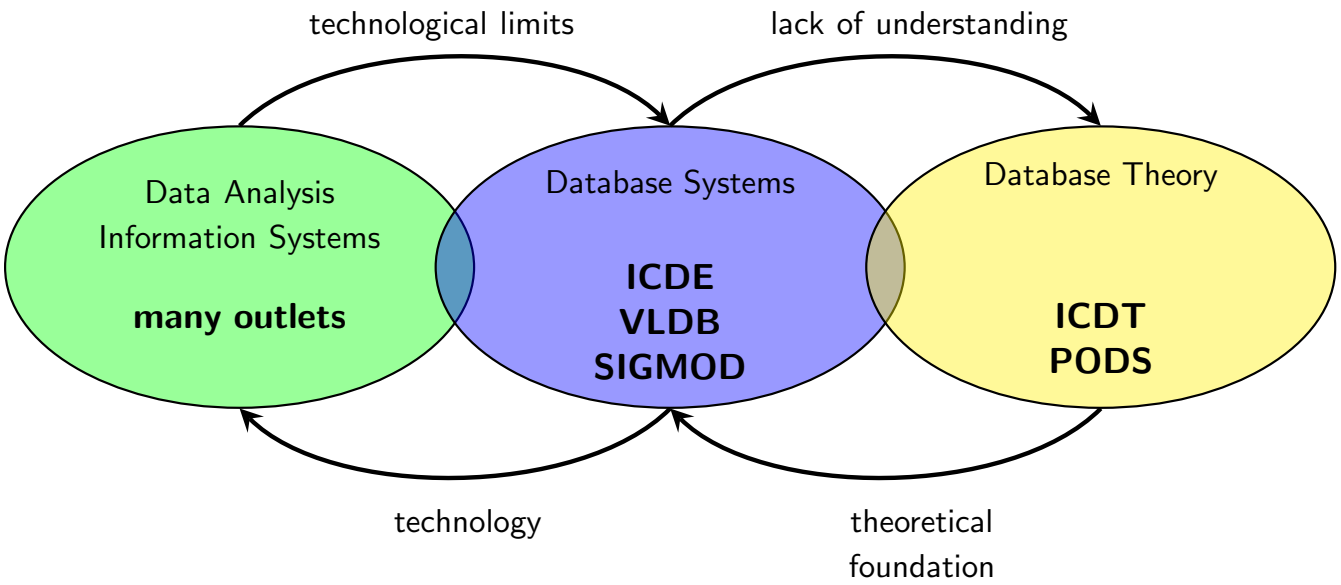
Overlap	Nested Loop Join	Prefix Join
0.95	2201 s	0.26 s
0.90	3136 s	0.44 s
0.80	4280 s	1.54 s
0.70	7770 s	4.93 s

Prefix Join 1500 to 8000 times faster than nested loop join!

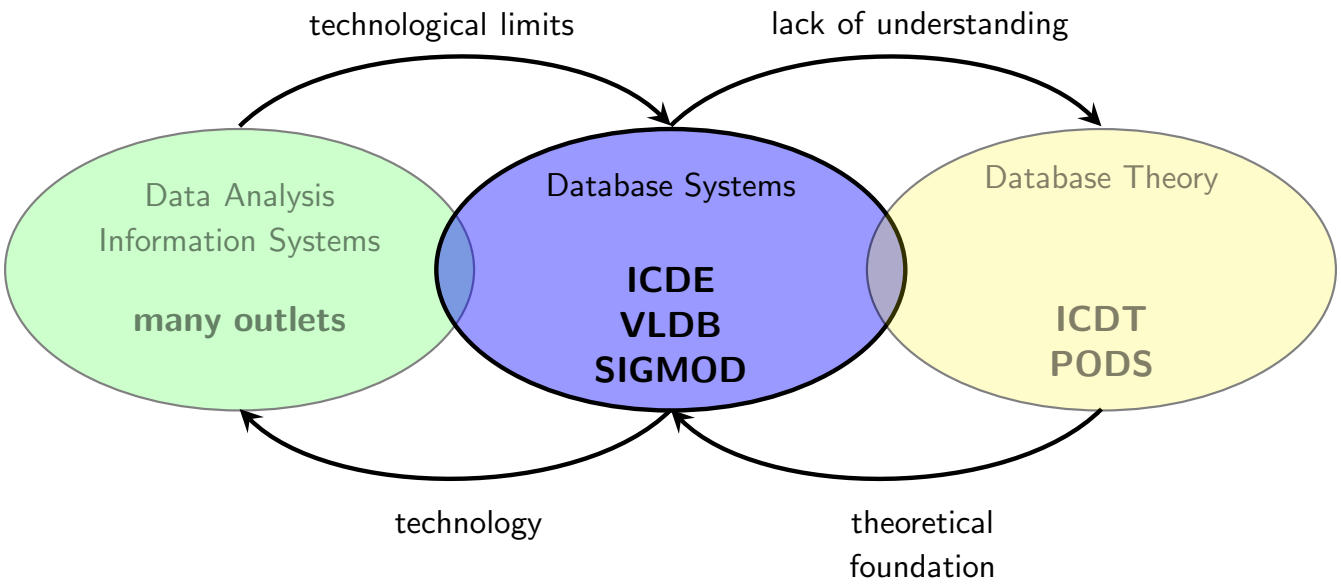
Summary

- Set similarity join
  - RTED: efficient tree edit distance
  - pq-grams: from trees to sets
  - prefix: avoiding nested loop join
- Publications
  - Very Large Database Conference (VLDB 2005, 2011, 2016)
  - ACM Trans. on Database Systems (TODS 2010, 2015)
  - Elsevier Information Systems (Inf. Syst. 2016)
- Algorithms and technology matter
  - efficient solutions orders of magnitude faster

Conclusion and Outlook

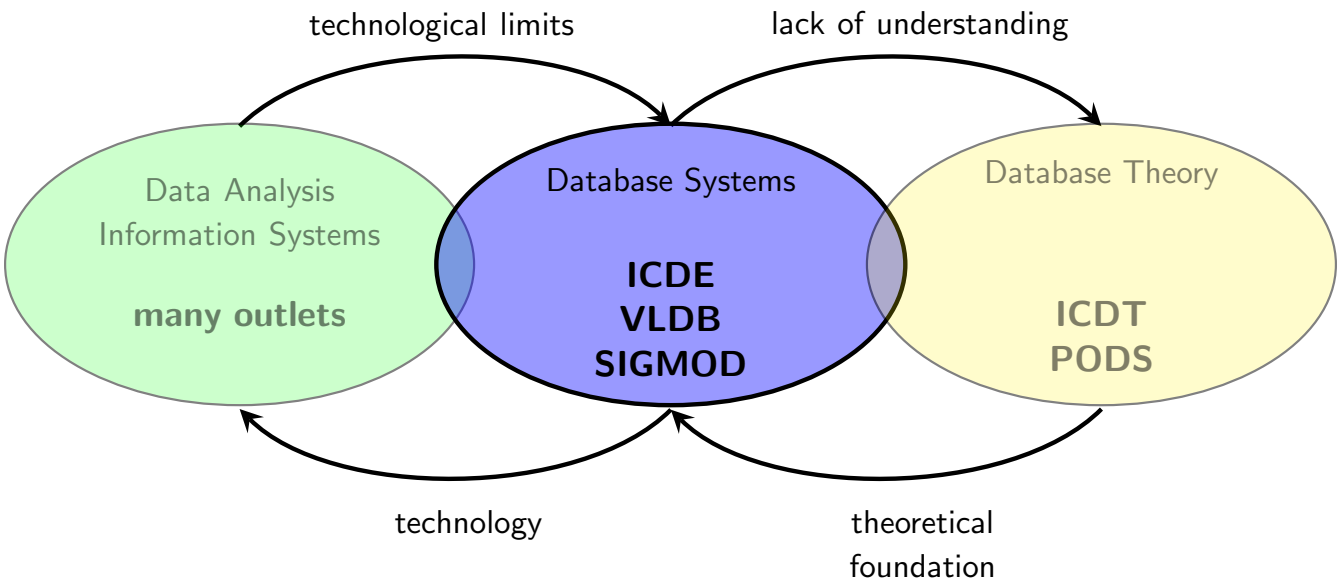


Conclusion and Outlook



Efficient technology is key enabler for innovation in big data.

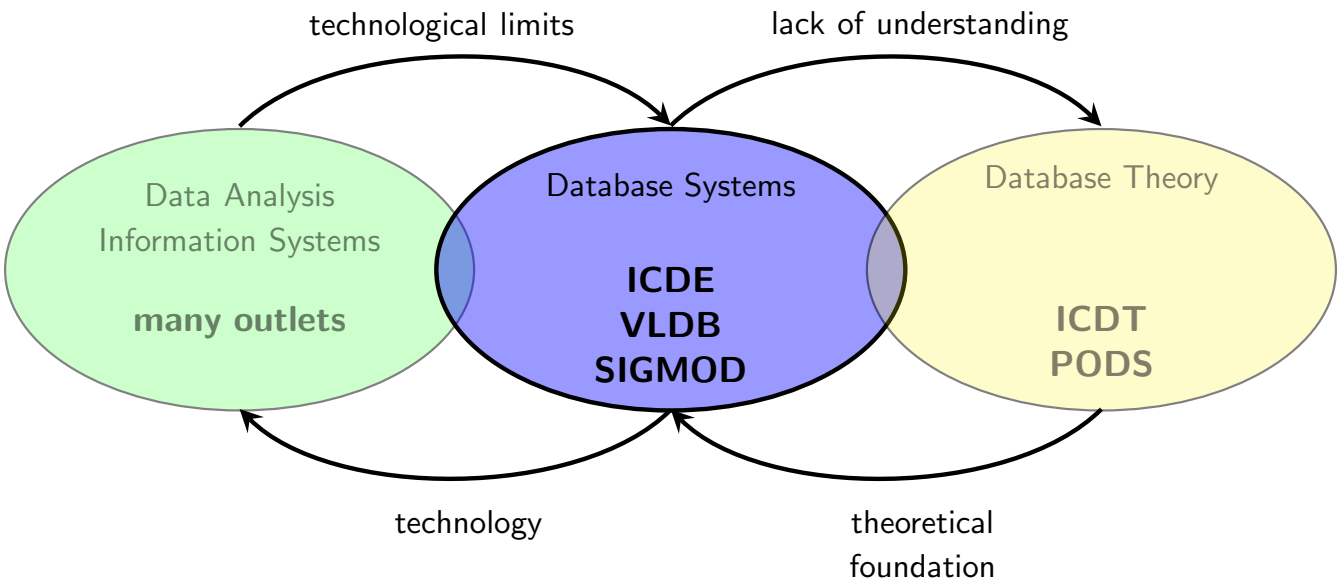
Conclusion and Outlook



- DACH publications in top-3 DBS outlets: only 1.5% from Austria

Efficient technology is key enabler for innovation in big data.

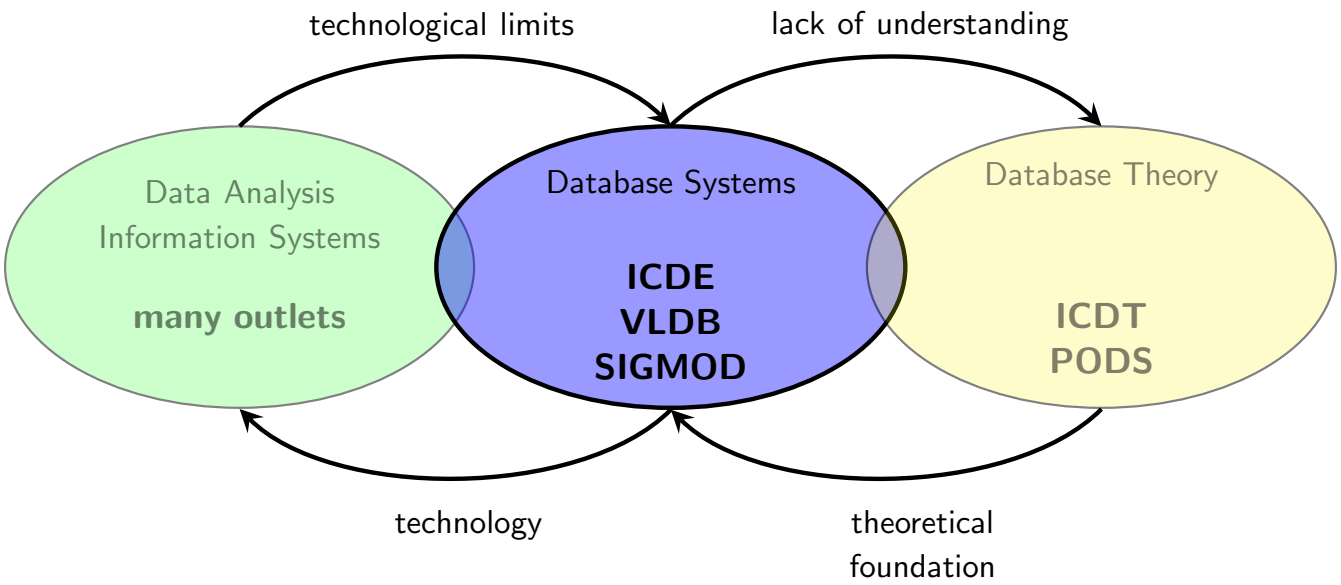
Conclusion and Outlook



- DACH publications in top-3 DBS outlets: only 1.5% from Austria
- Master Data Science in Salzburg:
  data analysis + applications + technology

Efficient technology is key enabler for innovation in big data.

Conclusion and Outlook



- DACH publications in top-3 DBS outlets: only 1.5% from Austria
- Master Data Science in Salzburg:
  data analysis + applications + technology
- Invitation: VLDB 2017 in Munich, EDBT 2018 in Vienna!

Efficient technology is key enabler for innovation in big data.

Thanks!

Thanks for your attention!

References I


Nikolaus Augsten, Michael Böhlen, and Johann Gamper.
Approximate matching of hierarchical data using *pq*-grams.
In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 301–312, Trondheim, Norway, September 2005. ACM Press.
ISBN 1-59593-154-6.


Nikolaus Augsten, Michael Böhlen, and Johann Gamper.
The *pq*-gram distance between ordered labeled trees.
ACM Transactions on Database Systems (TODS), 35(1):1–36, 2010.
ISSN 0362-5915.


Willi Mann, Nikolaus Augsten, and Panagiotis Bouros.
An empirical evaluation of set similarity join techniques.
PVLDB, 9(9):636–647, 2016.
ISSN 2150-8097.

References II


Mateusz Pawlik and Nikolaus Augsten.
RTED: A robust algorithm for the tree edit distance.
PVLDB, 5(4):334–345, December 2011.
ISSN 2150-8097.


Mateusz Pawlik and Nikolaus Augsten.
Efficient computation of the tree edit distance.
ACM Transactions on Database Systems (TODS), 40(1):3:1–3:40, March 2015.
ISSN 0362-5915.


Mateusz Pawlik and Nikolaus Augsten.
Tree edit distance: Robust and memory-efficient.
Information Systems, 56:157–173, 2016.
ISSN 0306-4379.